R/qtl2

rewrite of a very old R package

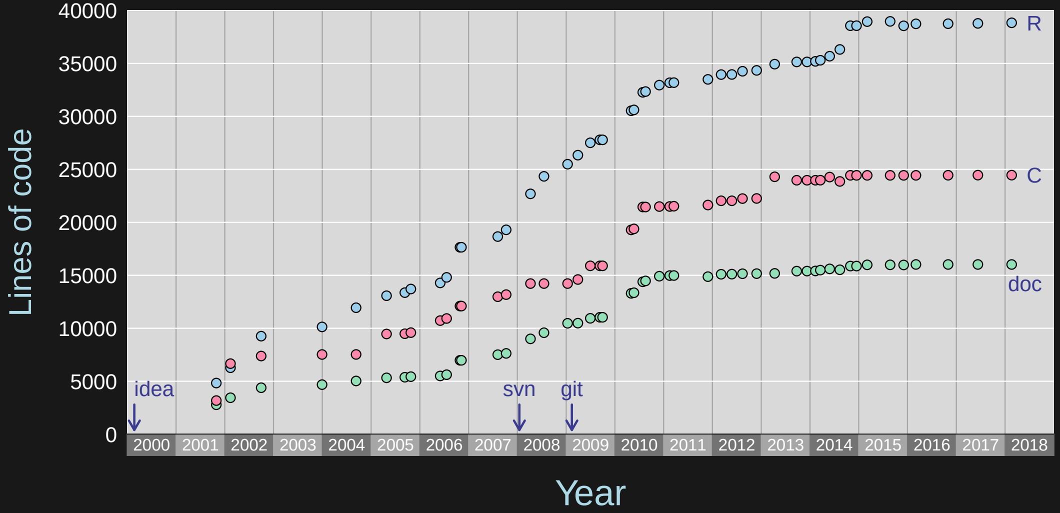
Karl Broman

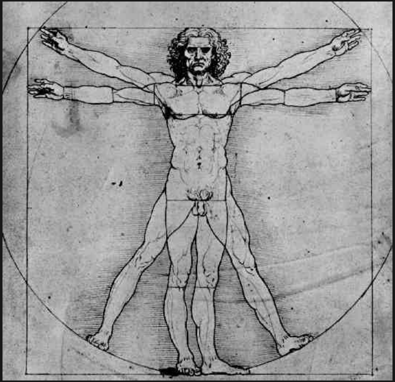
Biostatistics & Medical Informatics, U Wisconsin–Madison

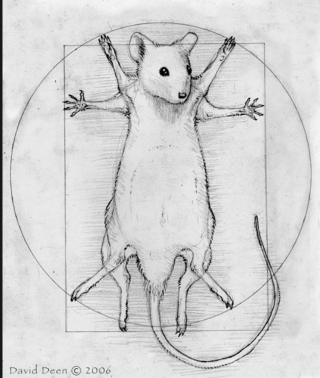
`kbroman.org`
`github.com/kbroman`
`@kwbroman`

Slides: `bit.ly/rstudio2019`

19 years of R/qtl

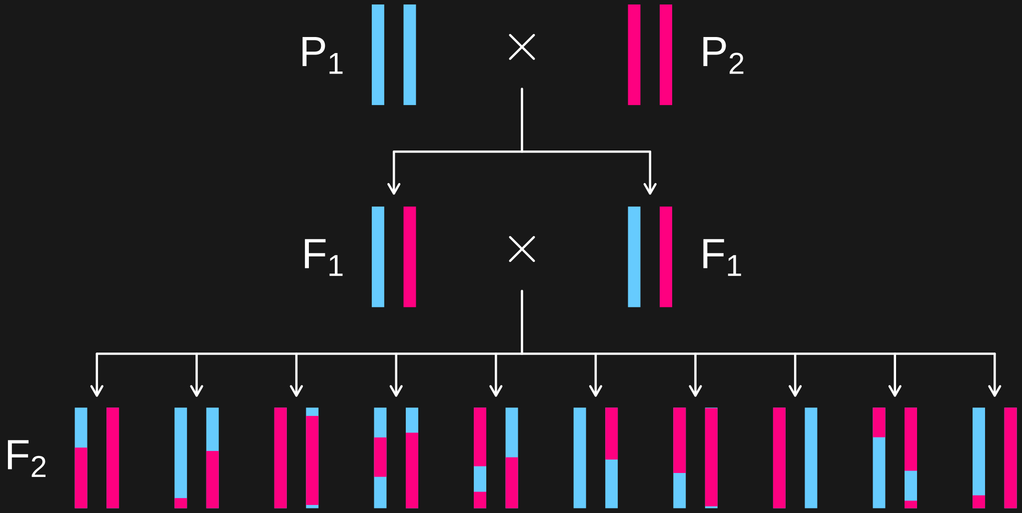




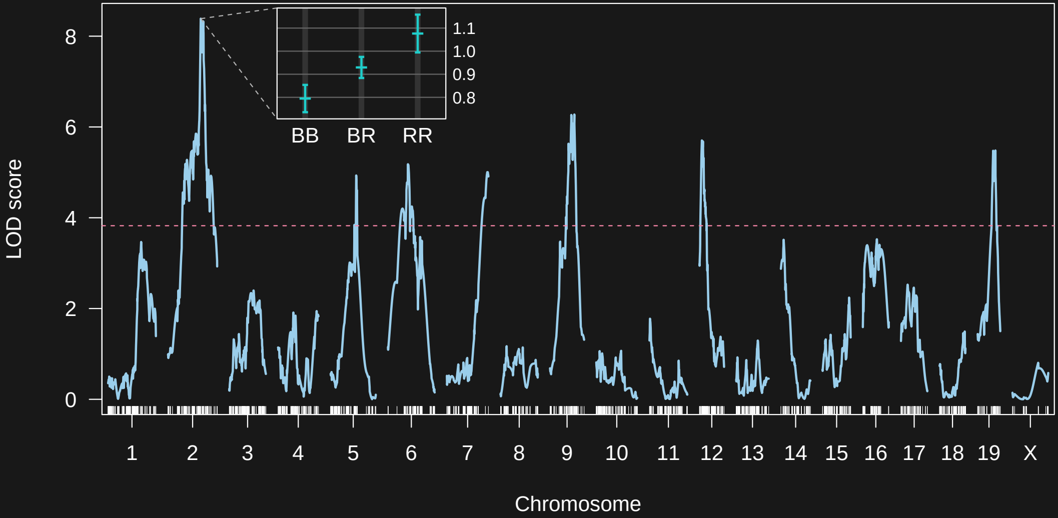


daviddeen.com

Intercross

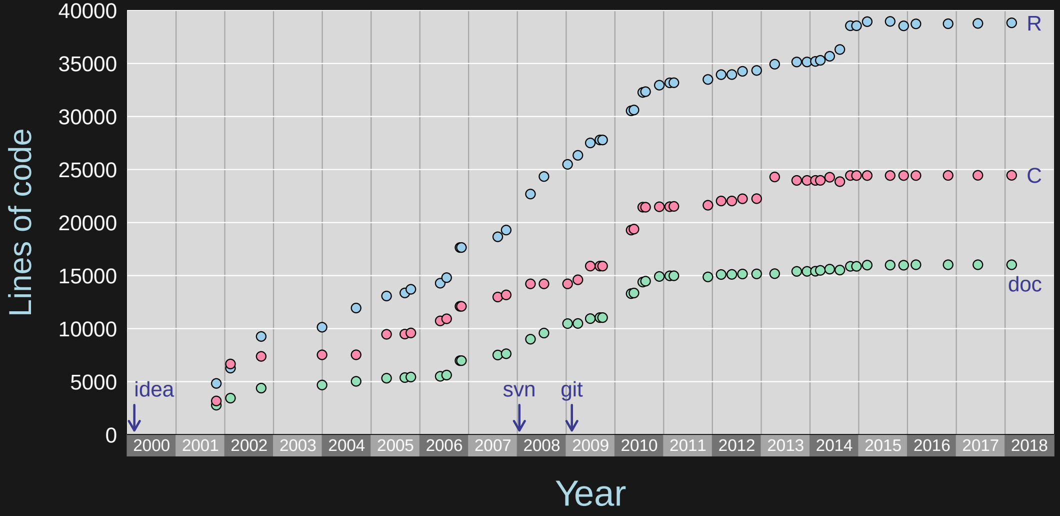


QTL mapping



LOD score
0
2
4
6
8
BB
BR
RR
1.1
1.0
0.9
0.8
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
X
Chromosome

19 years of R/qtl



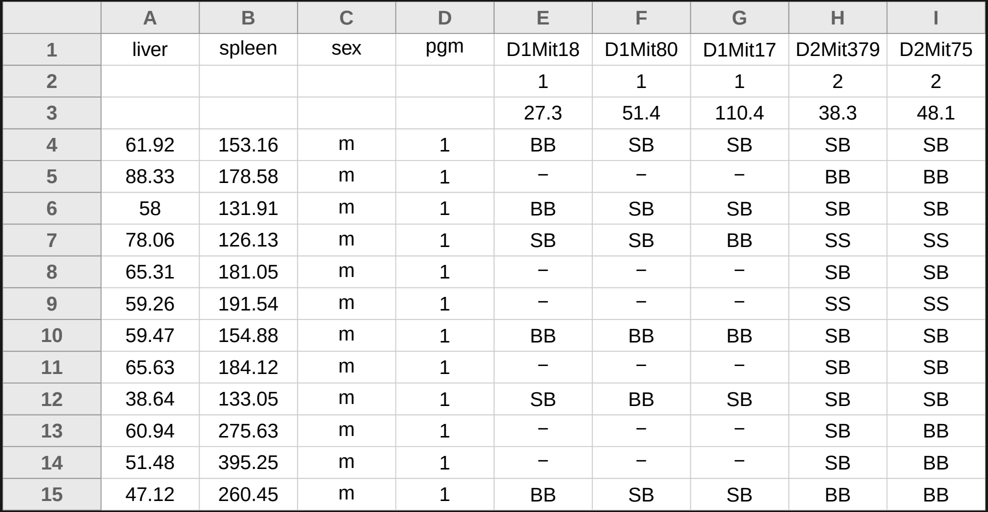
Good things

Good things

- some of the code
- basics of the user interface
- diagnostics and data visualization
- quite comprehensive
- quite flexible

Bad things

Input file



	A	B	C	D	E	F	G	H	I
1	liver	spleen	sex	pgm	D1Mit18	D1Mit80	D1Mit17	D2Mit379	D2Mit75
2					1	1	1	2	2
3					27.3	51.4	110.4	38.3	48.1
4	61.92	153.16	m	1	BB	SB	SB	SB	SB
5	88.33	178.58	m	1	–	–	–	BB	BB
6	58	131.91	m	1	BB	SB	SB	SB	SB
7	78.06	126.13	m	1	SB	SB	BB	SS	SS
8	65.31	181.05	m	1	–	–	–	SB	SB
9	59.26	191.54	m	1	–	–	–	SS	SS
10	59.47	154.88	m	1	BB	BB	BB	SB	SB
11	65.63	184.12	m	1	–	–	–	SB	SB
12	38.64	133.05	m	1	SB	BB	SB	SB	SB
13	60.94	275.63	m	1	–	–	–	SB	BB
14	51.48	395.25	m	1	–	–	–	SB	BB
15	47.12	260.45	m	1	BB	SB	SB	BB	BB

Stupidest code ever

```r
n <- ncol(data)
temp <- rep(FALSE,n)
for(i in 1:n) {
  temp[i] <- all(data[2,1:i]=="")
  if(!temp[i]) break
}
if(!any(temp)) stop("...")
n.phe <- max((1:n)[temp])
```

kbroman.org/blog/2011/08/17/the-stupidest-r-code-ever

Open source means
everyone can see my stupid mistakes

Open source means
everyone can see my stupid mistakes

Version control means
everyone can see every stupid mistake I’ve ever made

More typically bad code

The `scantwo()` function is 1446 lines long.

The related C code is 20% of the C code in R/qtl.

Baroque data structures

```
attr(mycross$geno[["X"]]$probs, "map")
```

Inconsistencies

```
plotPXG(mycross, marker="D5M250")

effectplot(mycross, mname1="D5M250")
```

Useless warning messages

```
X'X is singular.
```

Tests

> I don’t need tests; I have users.
>
> – Me

> If you use software that lacks automated tests,
> you are the tests.
>
> – Jenny Bryan

> It’s not that we don’t test our code,
> it’s that we don’t store our tests
> so they can be re-run automatically.
>
> – Hadley Wickham

How did this happen?

- I didn’t know any better
- Not enough re-writing
- Not enough planning
- Personal weaknesses
- Trade-offs (e.g. between interface and data structures)

Documentation

Tailored tutorials >

User guide >

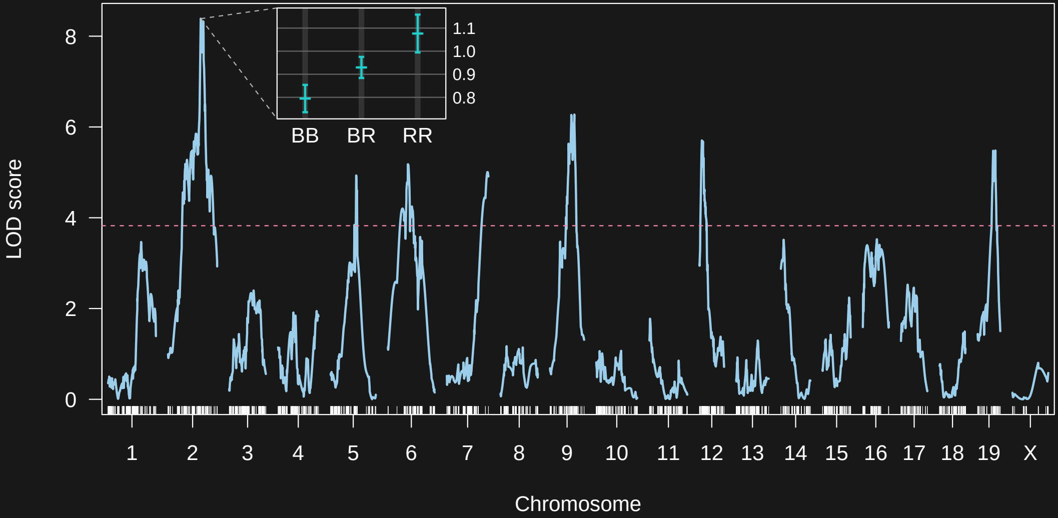
Examples >

Formal documentation

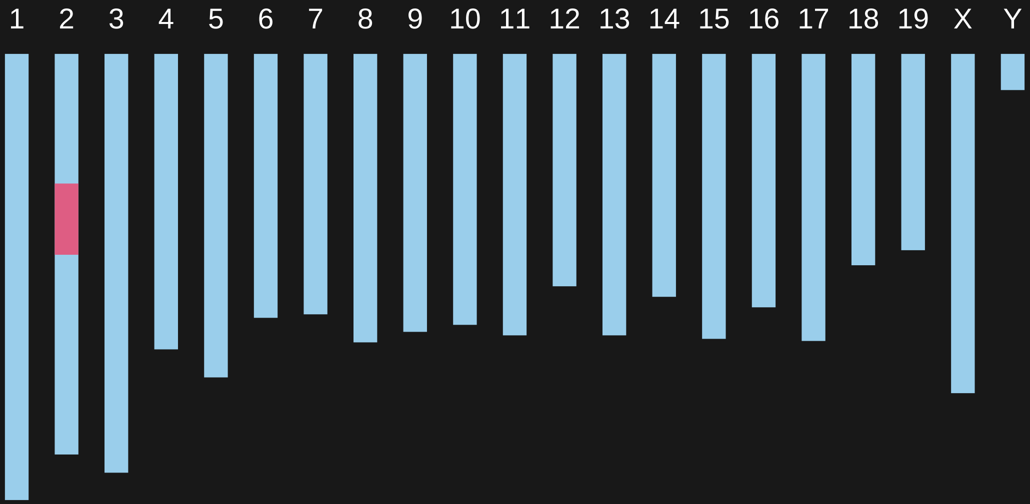
User support

- No question is too insignificant
- Treat each question like it is the first
- Take a breath before responding
- “Send me your data, and I’ll try to figure out what’s going on.”

QTL mapping

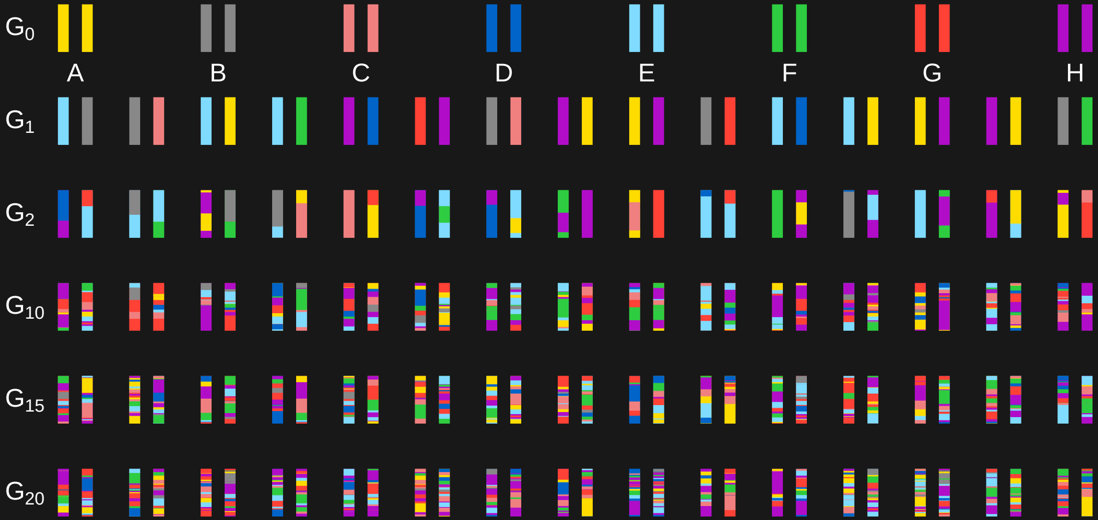

8
6
4
2
0
LOD score
1.1
1.0
0.9
0.8
BB
BR
RR
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
X
Chromosome

Congenic line



Advanced intercross lines

Heterogeneous stock



Genome-scale phenotypes

Alan Attie

ropenscilabs.github.io/miner_book

R/qtl2

- High-density genotypes
- High-dimensional phenotypes
- Multi-parent populations
- Linear mixed models

`kbroman.org/qtl2`

R/qtl2: Let’s not make the same mistakes

- C++ and Rcpp
- Roxygen2 for documentation
- Unit tests with testthat
- A single “switch” for cross type

R/qtl2: Let’s not make the same mistakes

- C++ and Rcpp
- Roxygen2 for documentation
- Unit tests with testthat
- A single “switch” for cross type
- Yet another data input format
- Flatter data structures, but still complex

Why devote such effort to R/qtl?

- For my own research
- So that others will use my methods ideas
- Has led to numerous collaborations
- I like to be useful

Acknowledgments

Danny Arends
Gary Churchill
Nick Furlotte
Dan Gatti
Ritsert Jansen
Pjotr Prins
Śaunak Sen
Petr Simecek
Artem Tarasov
Hao Wu
Brian Yandell

Robert Corty
Timothée Flutre
Lars Ronnegard
Rohan Shah
Laura Shannon
Quoc Tran
Aaron Wolen

NIH/NIGMS

R/qtl users

Slides: `bit.ly/rstudio2019`

`kbroman.org`

`kbroman.org/qtl2`

`github.com/kbroman`

`@kwbroman`